International Journal of Engineering and Applied Sciences (IJEAS)
ISSN: 2394-3661, Volume-3, Issue-9, September 2016

A New Approach for Finding Correlation Dimension Based on Escape Time Algorithm

Dr.Arkan Jassim Mohammed, Noora Ahmed Mohammed

***Abstract*— There are many technique to approximate the dimension of fractal sets. A famous technique to approximate the fractal dimension is correlation dimension. In this paper, we propose a new approach to compute the correlation dimension of fractals generated by the Escape Time Algorithm (ETA) and computing the correlation function by Grassberger-Procaccia method, was implemented using the Matlab program. A log –log graph is plotted by matlab program, the result will be an approximate polynomial of degree one (straight line) that's fits the data in a least square method. The correlation dimension of fractal object is the slope of this straight line.**

***Index Terms*—ETA, geometry, GPA, IFS.**

I. INTRODUCTION

The fractal geometry is a kind of geometry different from Euclidean geometry. The main difference is in the notation of dimension. The Euclidean geometry deals with sets which exists in integer dimension while the fractal geometry deals with sets in integer dimension. Dimension is the number of degrees of freedom available for a movement in a space. In common usage, the dimension of an object is the measurements that define its shape and size. The dimension, in physic, an expression of the character of a derived quantity in relation to fundamental quantities, without regard to its numerical value, while in mathematics, the dimension measurements are used to quantify the space filling properties of set [5]. Fractal geometry is a useful way to describe and characterize complex shapes and surfaces. The idea of fractal geometry was originally derived by Mandelbrot in 1967 to describe self-similar geometric figures such as the Von Koch curve. Fractal dimension D is a key quantity in fractal geometry. The D value can be a non-integer and can be used as an indicator of the complexity of the curves and surfaces. For a self-similar figure, it can be decomposed into N small parts, where each part is a reduced copy of the original figure by ratio r. The D of a self-similar figure, it can be defied as $D = -\log(N) / \log(r)$. For curves and images that are not self-similar, there exist numerous empirical methods to compute D. Results from different estimators were found to differ from each other. This is partly due to the elusive definition of D, i.e., the Hausdorff-Besicovitch dimension [4]. There are many kinds of dimensions. The correlation dimension is one of the fractal dimension measurement because it permits non-integer values. In (1983) Grassbergr and Procaccia [7, 8] published an algorithm (GPA) for estimating the correlation dimension that has become very popular and has been widely used since.

The correlation dimension D_{cor} can be calculated in real time as the fractal generated by Escape Time Algorithm by using the distances between every pair of points in the attractor set of N number of points, it is so-called correlation function $C(\varepsilon)$. It is defined as the probability that two arbitrary points on the fractal are closer together than the sides of size ε of the cells which cover the fractal.

The paper is organized as follows: In section 2, some background material is included to assist readers less familiar with detailed to consider. Section 3 introduces the concepts needed to understand the correlation function for calculating the correlation dimension D_{cor} , In section 4, we presents the Escape Time Algorithm. In section 5, the proposed method to compute the correlation dimension of fractals generated by Escape Time Algorithm. Finally, the conclusions are drawn in section 6.

II. THEORETICAL BACKGROUND

This section presents an overview of the major concepts and results of fractals dimension and the Iterated Function System (IFS). A more detailed review of the topics in this section are as in [1,3,6]. Fractal is defined as the attractor of mutually recursive function called Iterated Function System (IFS). Such systems consist of sets of equations, which represent a rotation, translation and, scaling. Given a metric space (X, d), we can construct a new metric space $(\mathcal{H}(X), h)$, where $\mathcal{H}(X)$ is the collection of nonempty compact set of X , and h is the Hausdorff metric on defined by:
$h(A, B) = \max\{\mathrm{d(A, B), d(B, A)}\}$.

Definition 1 [9] A transformation $f: X \to X$ on a metric space (X, d) is said to be a contraction mapping if for some $0 < r < 1$ $d(f(x), f(y)) \le r d(x, y)$, for all $x, y \in X$.

Theorem 1 (*contraction mapping theorem*)[3]. Let $f: X \to X$ be a contraction mapping on a complete metric space (X, d). Then f admits a unique fixed point x^* in X and moreover for any point $x \in X$, the sequence $\{f^{[n]}(x) : n = 0, 1, \ldots\}$ converges to x^*, where $f^{[n]}$ denotes the n-fold composition of f.

Definition 2 [6]. An IFS $\{X, \omega_1, \omega_2, \ldots, \omega_n\}$ consists of a complete metric space (X, d) and a finite set of contractive transformation $\omega_n: X \to X$ with contractivity factor r_n , for $n = 1, 2, \ldots, N$. The contractivity factor for the IFS is the maximum r among $\{r_1, \ldots, r_n\}$. The attractor of the IFS is the unique fixed point in $\mathcal{H}(X)$ of the transformation $W: \mathcal{H}(X) \to \mathcal{H}(X)$, defined by $A = W(A) = \cup_{i=1}^{N} \omega_i(B)$ for any $B \in \mathcal{H}(X)$.

Definition 3 [6]. A transformation $f: X \to X$ on a metric space (X, d) is called *dynamical system*. It is denoted by

Dr.Arkan Jassim Mohammed, Mathematics, Al-Mustansiriya/ Sciences/ Baghdad, Iraq.
Noora Ahmed Mohammed, Mathematics , Al-Mustansiriya / Sciences / Baghdad, Iraq.

$\{X; f\}$. The *orbit* of a point $x \in X$ is the sequence $\{f^{[n]}(x)\}_{n=1}^{\infty}$.

Theorem 2 [6]. Let $\{X, \omega_1, \omega_2, \dots, \omega_n\}$be an IFS with attractor A. The IFS is totally disconnected if and only if $\omega_i(A) \cap \omega_j(A) = \emptyset$, for all $i, j \in \{1, 2, \dots, N\}$ with $i \neq j$.

Definition 4 [6]. Let $\{X; \omega_n, n = 1, 2, \dots, N\}$ be totally disconnected IFS with attractor A. The transformation $f: A \to A$, defined by $f(a) = \omega_1^{-1}(a)$, for $a \in \omega_n(A)$, is called the *associated shift transformation* on A. They dynamical system $\{A; f\}$ is called the *shift dynamical system* associated with IFS.

III. Correlation dimension

There are many ways to define the fractal dimension, but one of the numerically simplest and most widly used is the correlation dimension of Grassberger and Porcaccia (1983) [7,8]. The real interest of the correlation dimension is determining the dimensions of the fractal attractor. Ther are others methods of measuring dimension such that hausdorff dimension, the box- counting dimension but the correlation dimension has the advantage of being straight forwardly and quickly claculated, when only a small number of points is a vailable, and is overwhelmingly concord with other calculations dimension.

Definition 5: Let $\{x_i\}_{i=1}^{N}$ be a sequence of points in $\mathbb{R}^n$. The correlation function, $C(\varepsilon)$ is defined by:

$$C(\varepsilon) = \lim_{N\to\infty} \frac{1}{N^2} \times \{\text{number of pairs } (x_i, x_j) \text{ of points with } d(x_i, x_j) < \varepsilon\}$$

$$C(\varepsilon) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i,j}^{N} \theta(\varepsilon - \|x_i - x_j\|)$$

where $\theta(x)$ is the Heaviside step function (unit step function) which has a value of either 0 or 1 and may be defined as:

$$\theta(\varepsilon - \|x_i - x_j\|) = \begin{cases} 1, & 0 \le \theta(\varepsilon - \|x_i - x_j\|) \\ 0, & 0 > \theta(\varepsilon - \|x_i - x_j\|) \end{cases}$$

which acts as a counter of the number of pairs of points with Euclidean separation - distance between two points on the fractal , x_i and x_j. The multiplier $\frac{1}{N^2}$ is included to normalize the count by the number of pairs of points on the fractal. .

Grassberger and Procaccia [7, 8] and Barnsley [6] established that for small values of the separation distance ε, the correlation function $C(\varepsilon)$ has been found to follow a power law such that

$C(\varepsilon) \sim C\varepsilon^{-D_{cor}}$

where $C(\varepsilon)$ is the number of cells with the edge size ε necessary to cover all points of the fractal object and C is a positive constant. The "~" symbol in this expression is used to indicate that this is not an exact equality but is a scaling relation that expected to be valid for sufficiently large N and small ε. After taking logarithms of each side of the scaling relation and rearranging terms, so we define

$$D_{cor} = \lim_{\varepsilon\to 0} \frac{\log C(\varepsilon)}{\log(\varepsilon)}$$

The correlation dimension D_{cor} estimated using the least squares linear regression of $\log C(\varepsilon)$ and versus $\log(\varepsilon)$, then the slope of linear model represent D_{cor}.

IV. The Escape Time Algorithm [6]

In this section, we study fractals generated by means of Escape time algorithms (ETA). Such fractals are computed by repeatedly applying a transformation to a given initial point in the plane, the algorithm is based on the number of iterations necessary to determine whether the orbit sequence tends to infinity or not that is, an orbit diverges when its points grow, further apart without bounds. Escape time algorithm is numerical computer graphical experiment to compare the number of iterations for the orbits of different points to escape from ball of large radius, centered at the origin. A fractal can then defined as the set whose orbit does not diverge.

The Escape Time Algorithm applied to any dynamical system of the form $\{\mathbb{R}^2; f\}, \{\mathbb{C}; f\}$, or $\{\hat{\mathbb{C}}; f\}$. Consider a window $\mathcal{W}$ and a region $\mathcal{R}$, to which orbits of points in $\mathcal{W}$ might escape. The result will be a "picture" of $\mathcal{W}$, where in the pixel corresponding to the point z is colored according to the smallest value of the positive integer n, such that $f^{[n]}(z) \in \mathcal{R}$. Let (a, b) and (c, d) be the coordinates of the lower left corner and the upper right corner respectively of a closed , filled rectangle $\mathcal{W} \subset \mathbb{R}^2$.

Let an array of points in $\mathcal{W}$ defined by:

$$x_{p,q} = \left(a + p\frac{(c-a)}{M}, b + q\frac{(d-b)}{M}\right), \quad \text{for } p, q = 0, 1, \dots, M$$

where M is a positive integer.

Define: $\mathcal{R} = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 > r\}$, where r is a positive number

compute a finite set of points: $\{x_{p,q}, f(x_{p,q}), f^{[2]}(x_{p,q}) \dots, f^{[n]}(x_{p,q})\}$ belonging to the orbit of $x_{p,q} \in \mathcal{W}$, for each $p, q = 0, 1, \dots, M$. The total number of points computed on an orbit is at most N (where N is a positive integer). If the set of computed points of the orbit $x_{p,q}$ contain no a point in $\mathcal{R}$ when $n = N$, then the pixel corresponding to $x_{p,q}$ will be black color, and the computation passes to the next value of (p, q). Otherwise the pixel corresponding to $x_{p,q}$ is assigned a color indexed by the first integer n, such that $f^{[n]}(x_{p,q}) \in \mathcal{R}$, and then the computation passes to the next value of (p, q).

So, the fractal generated by calculating the orbit for each point in the plane (that is pixel on screen) and checking whether it diverges. In our work, the black and white image of the fractal generating by coloring a pixel white if it is the orbit diverges i.e., $f^{[n]}(x_{p,q}) \in \mathcal{R}$, for some $n \le N$, and black if it is orbit does not i.e., $f^{[n]}(x_{p,q}) \notin \mathcal{R}$ for all $n \le N$. So, we have a closed subset F of $\mathcal{W}$ constructed by the Escape Time Algorithm, defined by:

$$F = \{x_{p,q} \in \mathcal{W}, \text{such that } f^{[n]}(x_{p,q}) \notin \mathcal{R} \text{ for all } n \le N\}$$
$$= \{x_{p,q} \in \mathcal{W}, \text{such that } x_{p,q} \text{ is black point}\}.$$

The set F is called the Escape Time Fractal.

The Escape Time Algorithm [6]**:**

1. Given $\mathcal{W} \subseteq \mathbb{R}^2$, s.t $\mathcal{W} = \{(x, y) : a \le x \le c, b \le y \le d\}$, define the array of point in $\mathcal{W}$ by , $x_{p,q} = (a + p(c-a)/M, b + q(d-b)/M)$, $p, q = 1,2, \dots, M$, for any positive integer M.
2. Let $\mathcal{C}$ be a circle centered at the origin, the set $\mathcal{R}$ is defined such that, $\mathcal{R} = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 > r\}$, where r is sufficiently large number.
3. Let f be a complex function with the orbit of point $\{f^{[n]}(x_{p,q})\}_{n=0}^{\infty}$, where $(x_{p,q}) \in \mathcal{W}$.
4. Repeat, $\forall x_{p,q} \in \mathcal{W}$.

IF $\{f^{[n]}(x_{p,q})\}_{n=0}^{\infty} \in \mathcal{R}$, then $x_{p,q}$ is colored with the color indexed by n.
Else it is colored Black.
End IF.

5. Change all that is not black color to the white color, such that:

$$x_{p,q} = \begin{cases} White & \text{if } f^{[n]}(x_{p,q}) \in \mathcal{R} \text{ for some } n \leq N \\ Black & \text{if } f^{[n]}(x_{p,q}) \notin \mathcal{R} \text{ if for all } n \leq N \end{cases}$$

6. Then the set of escape time point A in $\mathcal{W}$ is defined as follows,

$F = \{x_{p,q} \in \mathcal{W}: f^{[n]}(x_{p,q}) \notin \mathcal{R} \text{ for all } n \leq N\}$

$F = \{x_{p,q} \in \mathcal{W}$, such that $x_{p,q}$ is black point$\}$.

Output: The set F is called the fractal constructed by Escape Time Algorithm.

To apply the Escape Time Algorithm for IFS we find a relationship between the dynamical system $\{\mathbb{R}^2, f\}$ and IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ which is the shift dynamical system $\{F, f\}$ associated with the IFS.

So, we must follow these steps:

1. Find a dynamical system$\{\mathbb{R}^2, f\}$ which is an extension of a shift dynamical system associated with the IFS, and which tends to transform points off the fractal of the IFS to new points that are further away from the fractal (this is always possible if the IFS is totally disconnected).
2. Apply the Escape Time Algorithm, with $\mathcal{R}$ and $\mathcal{W}$ chosen properly.

V. The Correlation Dimension of Fractal Generated by Escape Time Algorithm

In this section, we propose on approach algorithm to compute correlation dimension of fractals generated by Escape Time Algorithm by finding shift dynamical system associated with IFS and the correlation function has been investigated by Grassbger–Procaccia Algorithm[7,8].

A log-log graph of the correlation function $C(\varepsilon)$ versus the distances between every pair of points ε in the fractals constructed by Escape Time Algorithm is an approximate of the correlation dimension, i.e., the correlation dimension is deduced from the slope of the straight line scaling grid in a plot of $\log C(\varepsilon)$ versus $\log(\varepsilon)$. We presented and implemented in the Matlab program listed in Appendix for computing the correlation dimension of fractals generated by Escape Time Algorithm.

The Proposed Algorithm (1):

INPUT: **READ** m , v { m = number of regions, v = value }

FOR $i = 1$ **TO** m
READ $\omega_i^{-1}(x,y) = (g_i(x), h_i(y))$
ENDFOR $\{i\}$
READ a , b , c , d , r, M , numits
PROCESS: **FOR** $P = 1$ **TO** M
FOR $q = 1$ **TO** M
$x = a + (c - a) * p/M$
$y = b + (d - b) * q/M$
FOR n=1 **TO** nmits
IF (x {≤, ≥, <, >} v) and (y{≤, ≥, <, >} v) **THEN**
$x = g_i(x)$, $y = h_i(y)$
ENDIF
IF $x * x + y * y > r$ **THEN**
Graph(p, q) , Color (n), n = numits
ENDFOR $\{n\}$
ENDFOR $\{q\}$
ENDFOR $\{p\}$
OUTPUT: GRAPH of m Regions
END.

{- - - - - -- - - - - - - Calculate correlation dimension (CorD) - - - - - - - - - - - - - - - - - -}

Step1: INPUT Nt, N {Nt=No. of transients, N=NO. of points}

Step2: Generate random $x_i, y_i, i = 1{:}Nt$

Step3: $d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, $mn_\varepsilon = \min(d)$, $mx_\varepsilon = \max(d)$

Step4: $T = \frac{2\log(mx_\varepsilon)}{\log(2)}$, $n_\varepsilon = \left[\frac{\log\left(\frac{T}{mn_\varepsilon}\right)}{\log(2)}\right] + 1$

Step5: $\varepsilon v_i = 2(mx_\varepsilon)\ (-i-1), i = 1, \dots, n_\varepsilon$

Step6: $N_p = \frac{N(N-1)}{2}$

$N_j = d < \varepsilon v_i$, $d > 0]$, $S = \sum N_j$, $c\varepsilon_i$ = concate $[c\varepsilon_i, S/N_p]$, $i = 1{:}n_\varepsilon$

Step7: $O_p = 3, k_1 = O_p + 1$, $k_2 = n_\varepsilon - O_p$
$xd_i = \frac{\log(\varepsilon v_i)}{\log(2)}$, $yd_i = \frac{\log(cv_i)}{\log(2)}$,
$x_p = xd_i$, $y_p = yd_i$, $i = k_1{:}k_2$

Step8: $a_1 x + a_0 = \boldsymbol{Polyit}\,(x_p, y_p, 1)$, $\boldsymbol{CorD = a_1}$

OUTPUT: PRINT (CorD)

In the following, this algorithm have been applied to the examples of fractals generated by Escape Time Algorithm. That is, we calculated the correlation dimension of fractal generated by Escape Time Algorithm.

***Example* 1:** Consider the IFS $\{\mathbb{R}^2, \omega_1, \omega_2, \omega_3\}$, where $\omega_1(x,y) = \left(\frac{1}{2}x, \frac{1}{2}y + \frac{1}{2}\right)$, $\omega_2(x,y) = \left(\frac{1}{2}x + \frac{1}{2}, \frac{1}{2}y\right)$, $\omega_3(x,y) = \left(\frac{1}{2}x, \frac{1}{2}\ y\right)$

The attractor of the IFS is the fractal Sierpinski triangle S. The relationship between the dynamical system $\{\mathbb{R}^2, f\}$ and fractals IFS $\{\mathbb{R}^2, \omega_1, \omega_2, \omega_3\}$ is a shift dynamical system $\{S, f\}$ associated with IFS. Calculate the inverse transformations for the above system, which are as follows:

Define $f : \mathbb{R}^2 \to \mathbb{R}^2$, by

$$f(x,y) = \begin{cases} \omega_1^{-1}(x,y) = (2x, 2y - 1) & if\ y > 1/2 \\ \omega_2^{-1}(x,y) = (2x - 1, 2y) & if\ x > 1/2\ and\ y \leq 1/2 \\ \omega_3^{-1}(x,y) = (2x, 2y) & otherwis \end{cases}$$

By applying Algorithm (1) to this dynamical system $\{\mathbb{R}^2, f\}$ with $\mathcal{W} = \{(x,y) \in \mathbb{R}^2 : a \leq x \leq c, b \leq y \leq d\}$, where $(a,b) = (0,0)$ and $(c,d) = (1,1)$ and $\mathcal{V} = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 > 2\}$, $f : \mathbb{R}^2 \to \mathbb{R}^2$, and N is larger enough we get the fractal set S constructed by this algorithm defined by : $S = \{(x,y) \in \mathcal{W} : f^{[n]}(x,y) \notin \mathcal{R} \text{ for all } n \leq N\}$.

That is, the black point represents the set S. See figure (1.a) and we find the correlation dimension by this figure (1.b) show the plot of $\log C(\varepsilon)$ against $\log(\varepsilon)$ for the fractal.

correlation dimension is computed from the sloped linear portion of the plot determined from a least square fit.

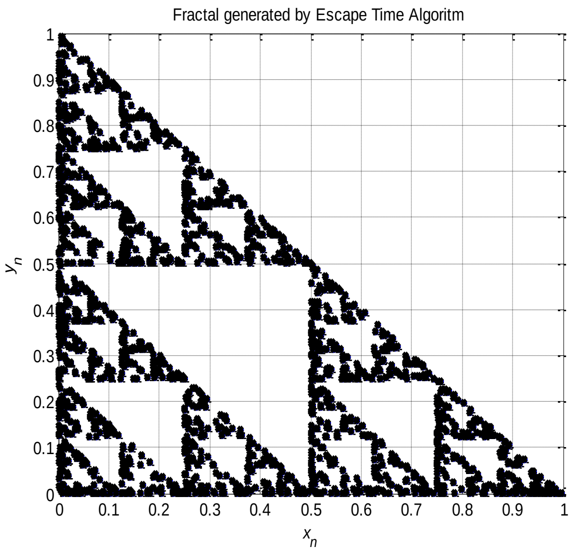

Figure (1.a) The Sierpinski triangle

In the following log-log graph indicate the correlation dimension $(D_{cor}(S) = 1.5868)$ of the Sierpinski triangle generated by a computer Matlab program

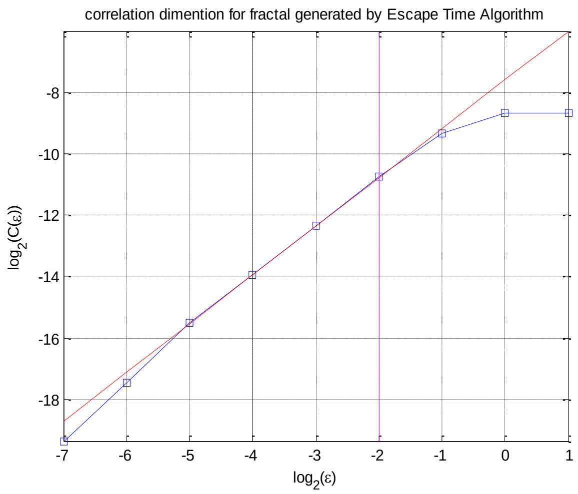

Figure (1.b)

Example 2: Consider the IFS $\{\mathbb{R}^2, \omega_1, \omega_2\}$, where $\omega_1(x,y) = \left(\frac{1}{3}x, \frac{1}{3}y\right)$, and $\omega_2(x,y) = \left(\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y\right)$

The attractor of the IFS is the fractal Cantor set C. The inverse transformations for the above system, which are as follows:

Define $f : \mathbb{R}^2 \to \mathbb{R}^2$, by

$$f(x,y) = \begin{cases} \omega_1^{-1}(x,y) = (3x, 3y) & if\ x \le \frac{1}{2} \\ \omega_2^{-1}(x,y) = (3x - 2, 3y) & if\ x > \frac{1}{2} \end{cases}$$

In the same manner as in example (1), we calculate the correlation dimension of cantor set generated by Escape Time Algorithm, $(D_{cor}(C) = 0.6351)$.

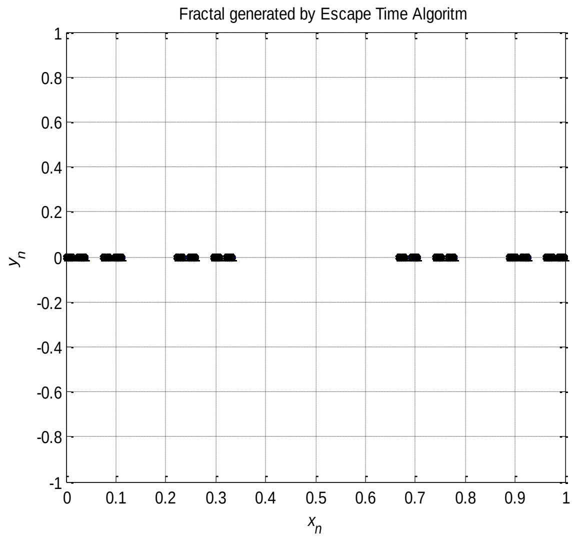

Figure (2.a): The Cantor set

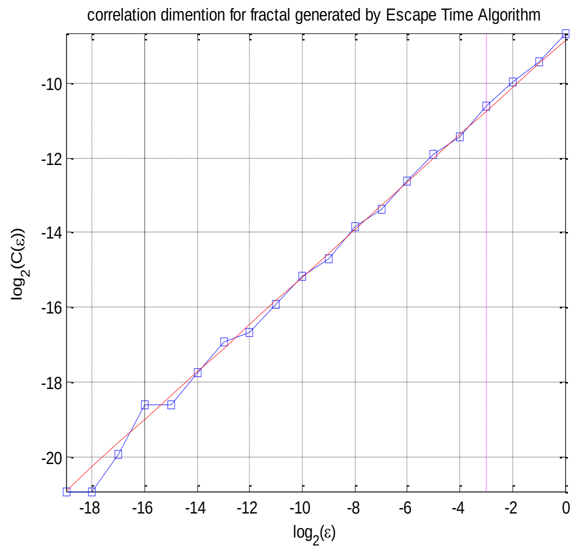

Figure (2.b)

Example 3: Consider the IFS $\{\mathbb{R}^2, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$, where $\omega_1(x,y) = \left(\frac{1}{4}x, \frac{1}{4}y\right), \omega_2(x,y) = \left(\frac{1}{4}x + \frac{3}{4}, \frac{1}{4}y\right), \omega_3\ (x,y) \left(\frac{1}{2}x + \frac{1}{4}, \frac{1}{2}y + \frac{1}{4}\right), \omega_4(x,y) = \left(\frac{1}{4}x, \frac{1}{4}y + \frac{3}{4}\right), \omega_5(x,y) = \left(\frac{1}{4}x + \frac{3}{4}, \frac{1}{4}y + \frac{3}{4}\right)$

The attractor of the IFS is the fractal T. The inverse transformations for the above system, which are as follows:

Define $f : \mathbb{R}^2 \to \mathbb{R}^2$, by

$$f(x,y) = \begin{cases} \omega_1^{-1}(x,y) = (4x, 4y) & if\ x \le \frac{1}{4}\ , y \le \frac{1}{4} \\ \omega_2^{-1}(x,y) = (4x - 3, 4y) & if\ x > \frac{1}{4}\ , y \le \frac{1}{4} \\ \omega_3^{-1}(x,y) = \left(2x - \frac{1}{2}, 2y - \frac{1}{2}\right) & Otherwise \\ \omega_4^{-1}(x,y) = (4x, 4y - 3) & if\ x \le \frac{1}{4},\ y \ge \frac{3}{4} \\ \omega_5^{-1}(x,y) = (4x - 3, 4y - 3) & if\ x > \frac{1}{4},\ y \ge \frac{3}{4} \end{cases}$$

In the same manner as in example (1), we calculate the correlation dimension of the fractal T generated by Escape Time Algorithm, ($D_{cor}(T) = 1.4384$).

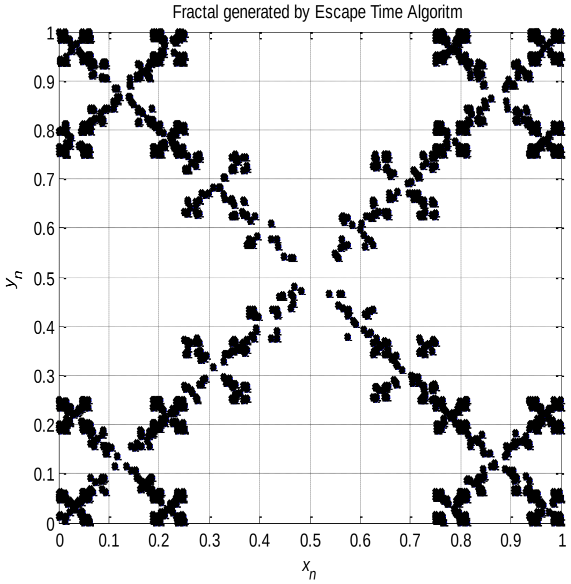

Figure (3.a)

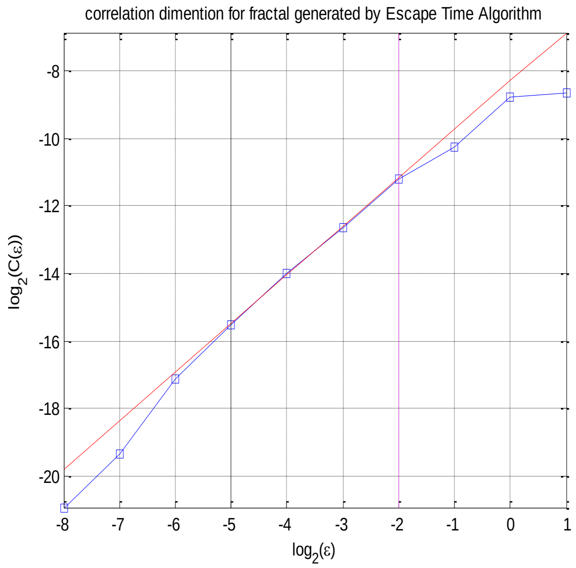

Figure (3.b)

***Example* 4:** Consider the IFS $\{\mathbb{R}^2, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8\}$, where

$\omega_1(x,y) = \left(\frac{1}{3}x, \frac{1}{3}y\right)$, $\omega_2(x,y) = \left(\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y\right)$, $\omega_3(x,y) = \left(\frac{1}{3}x, \frac{1}{3}y + \frac{2}{3}\right)$,
$\omega_4(x,y) = \left(\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y + \frac{2}{3}\right)$, $\omega_5(x,y) = \left(\frac{1}{3}x, \frac{1}{3}y + \frac{1}{3}\right)$,
$\omega_6(x,y) = \left(\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y + \frac{1}{3}\right)$, $\omega_7(x,y) = \left(\frac{1}{3}x + \frac{1}{3}, \frac{1}{3}y + \frac{2}{3}\right)$, $\omega_8(x,y) = \left(\frac{1}{3}x + \frac{1}{3}, \frac{1}{3}y\right)$

The attractor of the IFS is the fractal Sierpinski Carpet Sc. The inverse transformations for the above system, which are as follows:
Define $f : \mathbb{R}^2 \to \mathbb{R}^2$, by

$$f(x,y) = \begin{cases} \omega_1^{-1}(x,y) = (3x -, 3y) & if\ x \le \frac{1}{3}, y \le \frac{1}{3} \\ \omega_2^{-1}(x,y) = (3x-2, 3y) & if\ x \ge \frac{2}{3}, y \le \frac{1}{3} \\ \omega_3^{-1}(x,y) = (3x, 3y-2) & if\ x \le \frac{1}{3}, y \ge \frac{2}{3} \\ \omega_4^{-1}(x,y) = (3x-2, 3y-2) & if\ x \ge \frac{2}{3}, y \ge \frac{2}{3} \\ \omega_5^{-1}(x,y) = (3x, 3y-1) & if\ x \le \frac{1}{3}, \frac{1}{3} \le y \le \frac{2}{3} \\ \omega_6^{-1}(x,y) = (3x-2, 3y-1) & if\ x \ge \frac{2}{3}, \frac{1}{3} \le y \le \frac{2}{3} \\ \omega_7^{-1}(x,y) = (3x-1, 3y-2) & if\ \frac{1}{3} \le x \le \frac{2}{3}, y \ge \frac{2}{3} \\ \omega_8^{-1}(x,y) = (3x-1, 3y) & otherwise \end{cases}$$

In the same manner as in example (1), we calculate the correlation dimension of sierpinski carpet generated by Escape Time Algorithm, ($D_{cor}(Sc) = 1.8924$).

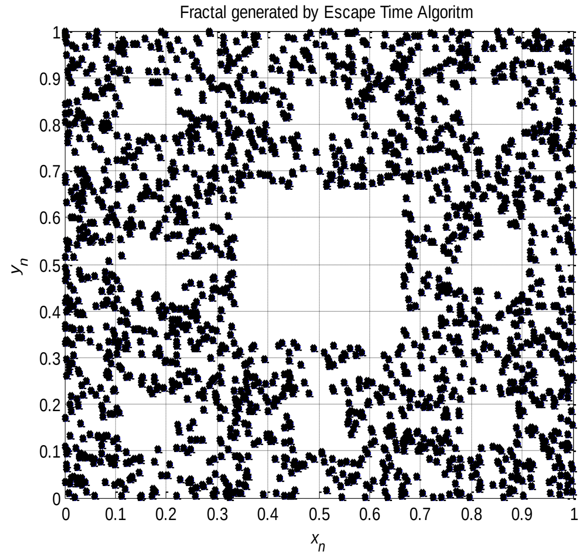

Figure (4.a): The Sierpinski Carpet

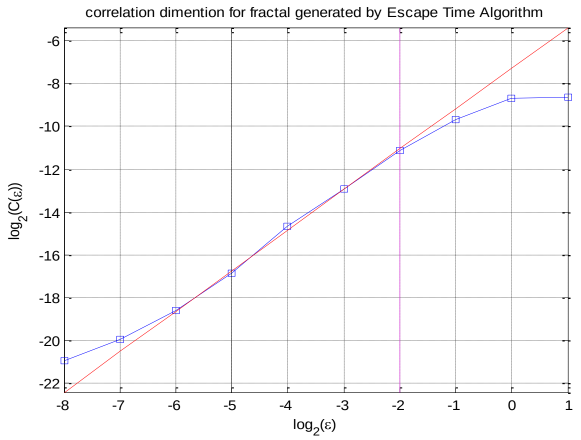

Figure (4.b)

***Example* 5:** Consider the IFS $\{\mathbb{R}^2, \omega_1, \omega_2\}$, where
$\omega_1(x,y) = \left(\frac{1}{2}x - \frac{3}{8}y + \frac{5}{16}, \frac{1}{2}x + \frac{3}{8}y + \frac{3}{16}\right)$,
$\omega_2(x,y) = \left(\frac{1}{2}x + \frac{3}{8}y + \frac{3}{16}, -\frac{1}{2}x + \frac{3}{8}y + \frac{11}{16}\right)$
The attractor of the IFS is the fractal G. The inverse transformations for the above system, which are as follows: Define

$$f:\mathbb{R}^2 \to \mathbb{R}^2, \text{by } f(x,y) = \begin{cases} \omega_1^{-1}(x,y) = \left(x+y-\frac{1}{2}, \frac{-4}{3}x+\frac{4}{3}y+\frac{1}{6}\right) & if\ x<\frac{1}{2} \\ \omega_2^{-1}(x,y) = \left(x-y+\frac{1}{2}, \frac{4}{3}x+\frac{4}{3}y-\frac{-7}{6}\right) & if\ x\geq\frac{1}{2} \end{cases}$$

In the same manner as in example (1), we calculate the correlation dimension of fractal G generated by Escape Time Algorithm. ($D_{cor}(G) = 1.2990$).

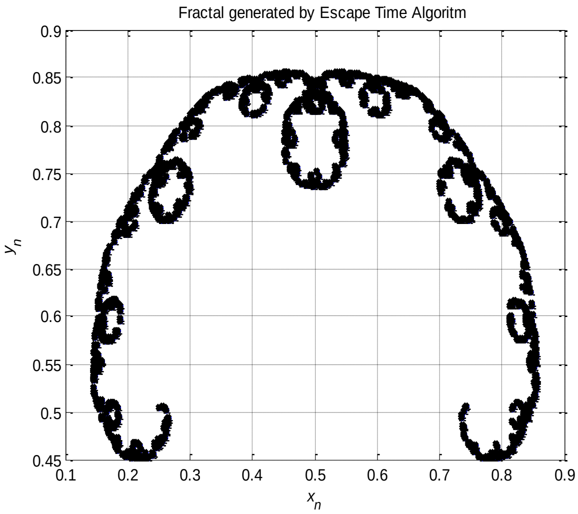

Figure (5.a)

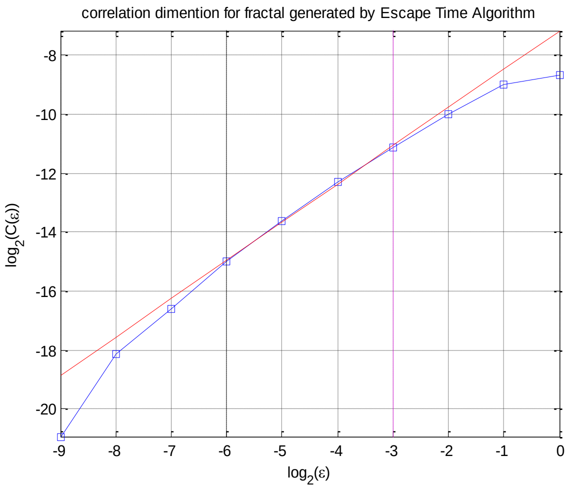

Figure (5.b)

VI. CONCLUSIONS

There are many approaches to compute the fractal dimension of an object. In this work we have presented an algorithm for computing correlation dimension D_{cor} of fractal generated by an Escape Time Algorithm (ETA) which is based on computing the correlation function. Computing the correlation function based on the selection of the Euclidean distance is presented and implemented the Matlab program listed in the Appendix for reducing the computational complexity of the Grassberger-procaccia Algorithm.

APPENDIX

Computer program for computing the correlation dimension of fractal generated by Escape Time Algorithm

{------------------------------Graph the Shapes using ETA ----------------------------------}

```
clc; clear all;
%a=1/3; b=0; c=0; d=1/3;
a=0; b=0; c=1; d=1;
numits = 4;  r = 1;
M=100;
A1=0; t=0; h=0;
method = '1';
for p= 1:M
   for q=1:M
      x = a + (c-a)*p/M;
      y = b + (d-b)*q/M;
      n = 0;
      while n <= numits
         n = n + 1;
         switch lower(method)
            case '1'
               %disp('Sierpinski Triangle')
               if (x>=0)&&(y>0.5)
                  x=2*x; y=2*y-1;
               end;
               if (x>0.5)&&(y<=0.5)
                  x=2*x-1; y=2*y;
               end;
               if (x<=0.5) && (y<=0.5)
                  x=2*x; y=2*y;
               end;
            case '2'
               %disp('Ex2')
               if (x<0.5)
                  x1=x+y-1/2; y1=4/3*y-4/3*x+1/6;
                  x=x1; y=y1;
               end;
               if (x>=0.5)
                  x1=x-y+1/2; y1=4/3*x+4/3*y-7/6;
                  x=x1; y=y1;
               end;
            case '3'
               %disp('Ex11')
               if (x<=1/4)
                  if (y<=1/4)
                  x=4*x; y=4*y;
                  end;
                  if (y>=3/4)
                     x=4*x; y=4*y-3;
                  end;
                  if (y>1/4)&&(y<3/4)
                     x=2*x-1/2; y=2*y-1/2;
                  end;
               else
                  if (y<=1/4)
                  x=4*x-3; y=4*y;
                  end;
                  if (y>=3/4)
                     x=4*x-3; y=4*y-3;
                  end;
                  if (y>1/4)&&(y<3/4)
                     x=2*x-1/2; y=2*y-1/2;
                  end;
               end;
            case '4'
               %disp('Cantor Set') Ex13
               if (x<=0.5)
                  x=3*x; y=3*y;
               else
                  x=3*x-2; y=3*y;
```

```
          end;
        case '5'
         %disp('Ex16'); %disp('Sierpinski Carpet')
          if (x<=1/3)
            if (y<=1/3)
            x=3*x; y=3*y;
            end;
            if (y>=2/3)
              x=3*x; y=3*y-2;
            end;
            if (y>1/3)&&(y<2/3)
              x=3*x; y=3*y-1;
            end;
          end;
          if x>=2/3
            if (y<=1/3)
            x=3*x-2; y=3*y;
            end;
            if (y>=2/3)
              x=3*x-2; y=3*y-2;
            end;
            if (y>1/3)&&(y<2/3)
              x=3*x-2; y=3*y-1;
            end;
          end;
          if (x>=1/3)&&(x<=2/3)
            if (y<=2/3)
            x=3*x-1; y=3*y;
            end;
            if (y>=2/3)
              x=3*x-1; y=3*y-2;
            end;
            if (y>1/3)&&(y<2/3)
              x=3*x-2; y=3*y-1;
            end;
          end;
        otherwise
          disp('Unknown method.')
      end
      if x^2 + y^2 > r
        t=t+1;
        P(t)=p; Q(t)=q;
        X(t)=x; Y(t)=y;
        n = numits+1;
        %fprintf('(%4.3f,%4.3f)=%4.3f\n',x,y,x+y);
      else
        h=h+1;
        P1(h)=p; Q1(h)=q;
      end;
      if (x^2 + y^2)^0.5 > R^2
        A1 = A1 + 1;
      end;
    end;
  end;
end;
figure(1)
plot(P(1:t),Q(1:t),'.black','MarkerSize',5);
xlabel('\itx_n')
ylabel('\ity_n')
title('    Fractal generated by Escape Time Algorithm ');
```

{-------------------------- Calculate correlation Dimensions ---------------------------------}

```
clear all;clc;
Ntrans=1000;
N_pts=3000;
x0=0; y0=0;
m=3;
a=[0.5 0 0 0.5 0 0;
  0.5 0 0 0.5 0.5 0;
  0.5 0 0 0.5 0 0.5];
px=y0; py=x0;
for j=1:Ntrans
  s=floor(m*rand+1);
  nx=a(s,1)*px+a(s,2)*py+a(s,5);
  ny=a(s,3)*px+a(s,4)*py+a(s,6);
  px=nx; py=ny;
end;
nx
ny
x=zeros(N_pts,1); y=zeros(N_pts,1);
x(1)=nx; y(1)=ny;
for j=1:N_pts-1
  s=floor(m*rand+1);
  x(j+1)=a(s,1)*x(j)+a(s,2)*y(j)+a(s,5);
  y(j+1)=a(s,3)*x(j)+a(s,4)*y(j)+a(s,6);
end;
x
y
figure(1);
axis tight
plot(x(1:N_pts),y(1:N_pts),'.b','MarkerSize',3);
xlabel('\itx_n')
ylabel('\ity_n')
title('    Fractal generated by Escape Time Algorithm');
grid
ED=sparse(N_pts,N_pts);
k=0;
for j=1:100 %N_pts
  for i=j+1:100 %N_pts
    d=(x(i)-x(j))^2+(y(i)-y(j))^2;
    ED(i,j)=d;
    k=k+1;
    if mod(k,250000)==0
    fprintf('%d - %3.2f\n',k,d);
    end;
  end;
end;
ED=sqrt(ED);
min_eps=double(min(min(ED+(1000*ED==0))));
m_eps=double(max(max(ED)));
max_eps=2^ceil(log(m_eps)/log(2));
n_div=floor(double(log(max_eps/min_eps)/log(2)));
n_eps=n_div+1;
eps_vec=max_eps*2.^(-((1:n_eps)'-1));
Npairs=N_pts*(N_pts-1)/2;
c_eps=[];
for i=1:n_eps
  eps=eps_vec(i);
  N = (ED<eps & ED>0);
  S =double(sum(sum(N)));
  c_eps = [c_eps; S/Npairs];
end;
omit_pts=3;
k1=omit_pts+1; k2=n_eps-omit_pts;
in_grid = k1:k2;
xd=log(eps_vec)/log(2);
```

```
yd=log(c_eps)/log(2);
xp=xd(in_grid); yp=yd(in_grid);
[coeff,temp]=polyfit(xp,yp,1);
D_Cor=coeff(1);
poly=D_Cor*xd+coeff(2);
D_Cor
figure(2);
plot(xd,yd,'s-');
hold on
plot(xd,poly,'r-');
axis tight
plot([xd(k1),xd(k1)],[-30,30],'m--');
plot([xd(k2),xd(k2)],[-30,30],'k--');
xlabel('log_2(\epsilon)');
ylabel('log_2(C(\epsilon))');
title({'correlation dimension for fractal generated by Escape
Time Algorithm'});
grid
```

REFERENCES

[1]. B. B. Mandelbrot, "The Fractal Geometry of Nature", W. H. Freeman and Co., , New York, NY 480. (1983).
[2]. D. Gulick, "Encounters with Chaos", McGraw-Hill, Inc., (1992).
[3]. G. A. Edgar, "Measure, Topology and Fractal Geometry", New York : Springer-Verlag, (1990).
[4]. G. Zhou, N.S.N. Lam, "A comparision of fractal dimension estimations based on multiple surface generation, computers and Geosciences, 31 (2005), 260-1269.
[5]. K. Falcone, "Fractal Geometry, Mathematical Foundations and Applications", John-Wiley and Sons Ltd, England (1999).
[6]. M. F. Barnsley, "Fractals. Every Where", Academic Press Professional, Inc., San Diego, CA, USA, (1993).
[7]. P. Grassberger, and I. Procaccia, "Characterization of strange attractors, Physical . Review Letters, 50 (1983a) , 346-349.
[8]. Grassberger, P, and Procaccia I., "Measuring the strangeness of strange attractors. Physica D, 9 (1983b), 189-208.
[9]. S. N. Elaydi, "Discrete Chaos", Champan and Hall/CRC, (2000)